

1. 실습 시간에 작성한 프로그램의 알고리즘과 자료구조를 요약하여 기술하시오.

1) class Array

new 연산자를 이용해서 int 타입의 1차원 배열을 동적 생성하고 배열의 인덱스를 나타내는 '[]'에 대해 연산자 다중정의를 한다. 이에 class Array로 생성된 객체에서도 사용이 가능하게 한다.

int *data	protected	배열 원소의 값을 저장한다.
int len	protected	배열의 크기를 저장한다.
Array(int size)	public	크기가 size인 배열을 생성한다.
~Array()	public	소멸자; 할당된 메모리를 해제한다.
int length() const	public	배열의 크기를 반환한다.
int& operator[] (int i)	public	배열의 i번째 위치의 주소를 반환한다.
int operator[](int i) const	public	배열의 i번째 위치의 값을 반환한다.
void print()	public	배열의 모든 내용을 출력해주는 함수다.

2) class RangeArray

class Array를 상속 받으며 index의 범위가 0부터 시작하는 것이 아닌, 생성 시 동적으로 정해진다. 예를 들어 RangeArray A가 (-10, 10)으로 생성되었다면 RangeArray는 index (-10, 10)을 갖는 것이다.

int low	protected	배열의 시작 index를 지정한다.
int high	protected	배열의 마지막 index를 지정한다.
int *data	protected	class Array에서 상속 받은 attribute
int len	protected	class Array에서 상속 받은 attribute
RangeArray(int low, int high)	public	Array 생성자를 상속 받아 크기가 high - low + 1인 배열을 생성한다.
~RangeArray()	public	~Array()가 실행되며 배열을 메모리에서 제거한다.
int length() const	public	class Array에서 상속 받은 method
int baseValue()	public	배열의 시작 index를 반환한다.
int endValue()	public	배열의 마지막 index를 반환한다.
int& operator[] (int i)	public	배열의 i - low번째 위치의 주소를 반환한다.
int operator[](int i) const	public	배열의 i - low번째 위치의 값을 반환한다.

2. 숙제 문제를 해결하기 위한 알고리즘 및 자료구조를 요약하여 기술하시오.

1) class Str

문자형 변수 str의 멤버함수를 구현한다. 문자열의 길이를 저장하고 값을 대입하기 위한 등호 연산자 및 문자열끼리의 값 비교가 가능한 함수를 구현한다.

char *str	protected	문자열의 내용을 저장한다.
int len	protected	문자열의 길이를 저장한다.
Str(int leng)	public	leng은 문자열의 길이이다. 만약, leng <= 0 이면 에러 메시지를 출력하고 길이를 0으로 지정한다.
Str(char *neyong)	public	neyong에는 초기화할 내용이 들어간다. 만약, neyong = NULL이면 에러 메시지를 출력하고 leng = 0 으로 한다.
~Str()	public	소멸자. str이 가리키는 메모리를 해제한다.
int length()	public	문자열의 길이를 반환한다.
char *contents	public	저장된 문자열의 내용을 반환한다.
int compare(char *a)	public	strcmp 함수로 str과 a를 비교한 결과를 반환한다. strcmp(Str, a)
int compare(Str &a)	public	strcmp 함수로 str과 a.contents()를 비교한 결과를 반환한다. strcmp(Str, a.contents())
void operator = (char *a)	public	기존 str에 할당된 메모리를 해제한다. str에 a의 내용을 대입한다.
void operator (class Str &a)	public	기존 str에 할당된 메모리를 해제한다. str에 a.contents()의 내용을 대입한다.

3. 아래 내용에 대한 추가 설명을 하시오.

1) OOP

Object-Oriented Programming, 객체 지향 프로그래밍의 뜻이며 프로그램을 구현하는 programming paradigm 중 하나이다.

2) OOP를 쓰는 이유

현실 세계의 문제를 해결하기 위해 길고 복잡한 코드는 비효율적이다. 이러한 문제를 해결하기 위해 OOP가 생겨났고 기본적으로 bottom-up 방식이며 객체들간의 유기적인 관계를 통해 간단하고 효율적인 프로그래밍이 가능하다.

3) 객체

객체는 말 그대로 대상을 나타내는 단어다.

4) 클래스

객체들이 공통적으로 갖는 속성들을 모아서 정의 내린 것을 클래스라고 한다. 객체와 클래스 간의 관계는 붕어빵과 붕어빵을 찍는 기계로 비유된다. 붕어빵을 찍어내는 기계는 붕어빵이라는 객체들을 생성하기 위한 틀을 제공한다. 즉, 붕어빵을 찍는 기계는 클래스를 의미하고 그 곳에서 만들어진 붕어빵은 각각의 객체가 되는 것이다.

5) 인스턴스

클래스를 기반으로 구현된 구체적인 실체를 의미한다. 즉, 어떠한 클래스를 구체화하여 인스턴스 되었다고 말한다.

6) 상속의 개념과 사용 이유

상속은 코드의 중복을 방지하기 위해 사용된다. 상속 관계를 맺으면 자식 객체 생성 시 부모 클래스의 속성들을 자동으로 상속 받기 때문에 같은 정의를 다시 할 필요가 없어진다.